Deep Dyna-Reinforcement Learning Based on Random Access Control in LEO Satellite IoT Networks

Xiangnan Liu, Haijun Zhang, *Senior Member, IEEE*, Keping Long, *Senior Member, IEEE*, Arumugam Nallanathan, *Fellow, IEEE*, and Victor C. M. Leung, *Fellow, IEEE*.

***Abstract*—Random access schemes in satellite Internet of Things (IoT) networks are being considered a key technology of new-type machine-to-machine (M2M) communications. However, the complicated situations and long-distance transmission can make the current random access schemes not suitable for the satellite IoT networks. The random access problem in the satellite IoT networks is studied in this paper. A novel random access scheme for machine-type-communication devices (MTCDs) is proposed, to maximize the efficiency of random access for contention-based and contention-free random access. Under the set of random access opportunities (RAOs) and limited delay, the random access control model is designed via maximizing efficiency of random access. The model-free deep reinforcement learning (DRL) algorithm is proposed to tackle the problem based on the random access model. Subsequently, the deep Dyna-Q learning algorithm is introduced to deal with the proposed random access control model. In this proposed scheme, the random access model-free DRL algorithm is developed using simulated experience. The proposed algorithms' performances are discussed, and simulation results show the desirable performance of the proposed DRL methods on different system parameters.**

***Index terms*—** Satellite IoT networks, deep Dyna-Q learning, efficiency of random access, MTCDs, RAOs.

I. Introduction

With the exponential growth of machine-type communication devices (MTCDs), machine-to-machine (M2M) communication has been widely utilized in various fields in the era of the Internet of Things (IoT) [1]. Traditional terrestrial networks involving cellular networks and WiFi networks have encountered challenges in capacity and data rate. Economic growth driven by M2M communication will rely on technological breakthroughs in existing communication systems to effectively relay large volumes of M2M traffic. If this is the case, the low earth orbit (LEO) satellite constellation system will serve as an indispensable slice for future mobile networks. The industry employs it for its shorter orbit altitude, lower transmission latency, and smaller path loss [2]. It forms a cellular service cell on the surface of the earth with a synthesis of LEO satellite constellation, the ground gateway, the network control center and user units. Covered by at least one LEO satellite, the MTCDs in the service area enable time-unlimited access to the system. In recent years, 3GPP has launched a research project on non-terrestrial networks, aiming to deploy satellite systems as standalone solutions to integrating with terrestrial cellular networks in MTC scenarios. Substantial projects by the European Space Agency have been in progress on enhancement of M2M and IoT technologies in satellite networks. Researchers, satellite operators, and Internet corporations have been contributing on the incorporation of the existing IoT developed protocols into satellite networks [3].

It is recommended that the random access technology to substantial low-power devices in the broadcast region be adopted. Realizing the signal transmission in the IoT networks via LEO satellites constellation. Random access technology is to allocate system resources, which plays a vital role in improving system resource utilization, reducing terminal usage, and saving terminal power consumption. Under the pretext of the existing protocol, the access class barring (ACB) framework is a typical random access protocol framework in MTC scenario [4]. In the ACB mechanism, when the terminal load exceeds the carried traffic, some terminals' access requests can be limited, and the physical random access channels' (PRACHs) load can be reduced. Likewise, the collision of the preamble sequences can be reduced accordingly. Underpinned by the ACB mechanism, a set of control parameters for a new type of terminal (or business) is configured while the levels of random access are not extended. The enhanced access barring (EAB) mechanism has been put on the agenda [5]. As an innovative concept

X. Liu, H. Zhang, and K. Long are with Beijing Engineering and Technology Research Center for Convergence Networks and Ubiquitous Services, University of Science and Technology Beijing, Beijing, China, 100083 (email: xiangnan.liu@xs.ustb.edu.cn, haijunzhang@ieee.org, longkeping@ustb.edu.cn).

Arumugam Nallanathan is with the School of Electronic Engineering and Computer Science, Queen Mary, University of London (e-mail: a.nallanathan@qmul.ac.uk).

Victor C. M. Leung is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, V6T 1Z4, Canada (e-mail: vleung@ece.ubc.ca).

states, the random access opportunity (RAO) represents the preamble sequences sent in the same random access time slot and frequency band [6]. RAO is the product of a sum total of preamble sequences and the available PRACHs. In terms of performance evaluation of random access, it has a wide range of applications in the current wireless communication system.

Due to a joint result of the long distance, increasing transmission rate and frequent mobility management, devices have to simultaneously access the network to preempt radio resources. On the one hand, the preamble sequences are limited, resulting in a lower access success rate and higher network congestion. On the other hand, PRACHs could be overburden with too many MTCDs to be reconnected [7]. Slotted ALOHA is applied to satellite communications, and this access method produces higher throughput [8]. In slotted ALOHA, the time axis is divided into several slots. MTCDs can send packets in the scheduled slots. When congestion happens, the MTCDs will resend packages through the ACK scheme. M. Bacco et al. introduced the idea of TCP into the satellite random access link to solve congestion control [9]. O. del Rio Herrero and R. De Gaudenzi analyzed the superiority of random access protocol in the satellite network, starting with the two technical perspectives of TDMA and CDMA [10]. As for real-time transmission, random access can boost the rate than before under the scheme of demand assignment multiple access. The CRDSA utilizing contention resolution can reduce the probability of contention and increase the probability of successful access. [11]. Additionally, CRSDA helps restore the packets in congestion by the means of successive interference cancellation, enhancing the total system's throughput to avoid preventable loss of packets. On a basis of CRDSA, Riccardo et al. studied an asynchronous contention resolution diversity ALOHA, which merely defines the practice and frame boundaries locally on the terminal. Each terminal is completely asynchronous [12]. In [13], the authors designed several ACB factors to increase the probability of successful access and decrease the access time delay. Prashant K et al. proposed a novel analytical model for the EAB mechanism to obtain its performance indicators. In addition, they constructed a corresponding energy consumption model to satisfy IoT PRACH requests [14].

A. Motivations and Contributions

In recent years, modeling and analyzing satellite IoT networks has been a hot topic in academic research. M. Bacco et al. explored the impact of M2M communication traffic under the condition of resource-constrained random access satellite links [15]. By using CRSDA, a closed-loop link congestion control was designed. The throughput estimation model facilitating satellite random access can accurately adapt to the satellite scenario's simulation [16]. As a promising technology, deep reinforcement learning (DRL) has also been proposed to be applied to congestion control in the satellite IoT networks [17]. The employment of DRL assists to attain better performance under the $\varepsilon-$ greedy algorithm. However, the modeling of random access in satellite networks, especially the LEO satellite constellation system, is little-researched. Lin et al. proposed a joint beamforming and power distribution algorithm for non-orthogonal multiple access in the air-space-ground integrated networks [18]. This iterative multiplication algorithm improves computational efficiency in contrast to the original one. With the regard to the case of minimizing the control channel flow in the LEO satellite communication system, Ivanov et al. proposed a mechanism of low-power resource allocation [19]. In [20], the authors focused on collecting data from geographically distributed IoT via LEO satellites. The method ensures queue stability, and maximizes the total amount of uploaded data while minimizing energy consumption. It suggests that a genetic algorithm for joint power and bandwidth allocation in a multi-beam satellite system exhibit greater flexibility in changing circumstances [21]. Premised on the current LTE system, the random access scheme in LEO satellite networks therefore is worthy of further study.

DRL is important in the field of machine learning, since it can effectively solve the high computing requirements brought by the vast state space and bolster the algorithm's capacity of learning in unknown environments [22]. At present, DRL has been commonly used in wireless communication network resource planning and other fields. Supported by the Markov decision process, incorporated with the function approximation of deep neural networks (DNNs), a strategy is yielded and thereby execute resource optimization. Reinforcement learning has obtained widespread application in dynamic multi-channel access. In dynamic multi-channel access, reinforcement learning has a wide range of applications. In [23], the author used a partially observed Markov process to solve the multi-channel access problem in wireless networks with the help of DQN. On this basis, a distributed multi-user reinforcement learning using DQN was considered on the issue of multi-spectrum access [24]. Subsequently, Zhong et al. used Actor-Critic's idea to design an algorithm to deal with the dynamic multi-channel access problem and compared it with the previous DQN method [25]. In [26], DRL was used for multiple access in heterogeneous wireless networks, which can achieve near-optimal performance. In terms of power allocation, Nasir et al. proposed a power allocation scheme based on model free DRL for a distributed execution dynamic system [27]. Based on previous research, Meng et al. considered cross-cell channel state information requests and proposed a dynamic kilometer allocation method for multi-agent DRL [28]. The

immediate-training concept of DRL is more suitable for various situations in LEO satellite networks random access.

There is limited research on the random access model of the LEO satellite IoT networks and the practical solution tool of DRL. Therefore, the proposal of utilizing the DRL method with innovative algorithms is feasible, especially in addressing the problem of random access in the LEO satellite IoT networks.

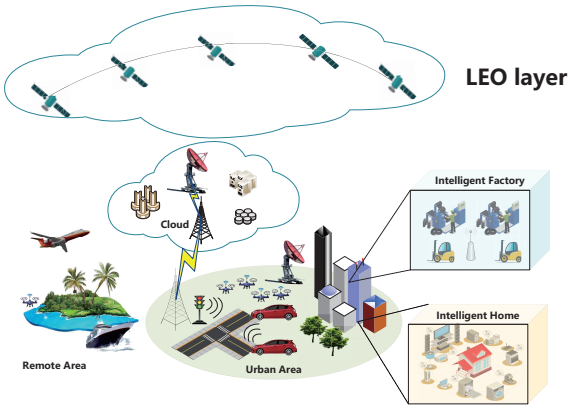

Fig. 1. LEO satellite IoT networks.

The present study is a preliminary attempt to explore the random access mechanism in the LEO constellation system satellite environment. It embraces the Markov decision process to characterize LEO satellites' current RAO deployment in several time slots, on a variety of machine types. It seeks to optimize access efficiency of RAO and EAB mechanisms. The DQN method in DRL is combined with the designed mechanism to maximize efficiency of random access, and it is compared with existing algorithms.

- Grounded on the Markov's decision process, a novel random access control mechanism for satellite IoT networks is initiated with an objective of model construction with the tool of state transition probability. The amount of access devices subject to LEO satellites is characterized by the elevation angle and motion trajectory at the moment.
- The framework of Dyna-DRL is the pivot of the proposed LEO satellite IoT networks' random access. Through Dyna-Q learning, RAOs are allocated to deliver the maximum efficiency of random access. The DNN is implemented a Q function for strategic planning.
- Two random access mechanisms are considered, Dyna mechanism and relatively low-complexity model-free mechanism. By comparison, the proposed deep Dyna-Q learning in random access control of higher efficiency.
- The proposed system identify the performance of contention-based and contention-free random access mechanisms on two different traffic models respectively. The proposed algorithm demonstrates a stable performance when serving both synchronous traffic and asynchronous traffic.

B. Organization

The rest of the paper is developed as follows. Section II presents the LEO satellite IoT networks random access system model and the problem formulation of the proposed random access system model via the Markov decision process. Section III reveals a deep Dyna reinforcement learning method to attain the efficiency of random access at its best in the LEO satellite IoT networks. Section IV highlights and compares the proposed algorithm with the method with relatively low computational complexity. In Section V, the performance of the proposed algorithm is verified by simulations. Finally, Section VI concludes this paper.

II. System Model and Problem Formulation

A. System Model

As shown in Fig. 1, the LEO satellite IoT random access networks constitute of $\mathcal{J}$ LEO satellites, and j refers to the jth LEO. Each LEO satellite will serve the MTCDs in its broadcast coverage and receive random access requests from massive MTCDs. Additionally, The PRACH uses slotted ALOHA as current available access method in LEO satellite networks. The number of RAOs available is the product of preamble sequences and PRACHs idle. The maximum preamble sequences available in each LEO is 54, and PARCHs compose of a discrete set $\mathbf{A} = \{0.5, 1, 2, 3, 5, 10\}$. So the allocated RAOs are $\{27, 54, 108, 176, 270, 540\}$. The maximum of allocated RAOs a_{max} is 540.

Firstly, the coverage time of the LEO satellite j should be considered [29]. The coverage time T_j between the satellite j and MTCDs is

$$T_j = \frac{2}{\omega_j} \arccos\left(\frac{\cos\gamma_{\max}}{\cos\gamma\left(t\right)}\right), \tag{1}$$

$\omega_j = \omega_{js} - \omega_e\beta_j$ is LEO satellite j's angular velocity. ω_{js} is the angular velocity in geocentric inertial coordinate system satellite j, ω_e is the angular velocity of the earth's auto rotation, and β_j is the orbital inclination angle. The $\gamma\left(t\right)$ is the geocentric angle between MTCDs and sub-satellite point of satellite at time t. The γ_{max} is the maximal geocentric angle.

MTCDs are randomly distributed on the ground, assuming that the sub-satellite points' distance follows a uniform distribution. Therefore, when the LEO satellite j covers ground

MTCDs, $\gamma(t)$ satisfies $U(0, \gamma_{max})$ uniform distribution. So $\gamma(t)$ can be derived from

$$f_{\gamma(t)}\left(\gamma\left(t\right)\right) = \begin{cases} 1/\gamma_{\max}, & 0 \leq \gamma\left(t\right) < \gamma_{\max} \\ 0, & others \end{cases}, \quad (2)$$

Given that the LEO satellite moves at a low orbit altitude and high speed, a quantity of MTCDs on the ground will realize random access over a specific coverage area of its characteristic time, and the maximum elevation angle $\theta_j^{\max}$ between the LEO satellite j and MTCDs it serves in a specific period is

$$\theta_j^{\max} = \arctan\left(\frac{\cos\alpha_j - r_e/(r_e + h_j)}{\sin\alpha_j}\right), \quad (3)$$

α_j is the latitude at projection of the LEO satellite j on the surface of the earth. r_e is the radius of the earth, which is set as 6300 km. $h_j \in$ [200 km, 2000 km] is the orbital altitude of LEO satellite j.

Then the inclination β_j of the LEO satellite j coverage is:

$$\beta_j = \begin{cases} \theta_j & \theta_j \leq \theta_j^{\max} \\ 2\theta_j^{\max} - \theta_j & \theta_j > \theta_j^{\max} \end{cases}, \quad (4)$$

The smaller the inclination angle β_j is, the longer the direct distance is. Similarly, the more coverage time it can last, the longer it takes to achieve access.

B. Problem Formulation

The Markov decision process underpins reinforcement learning. A typical Markov decision process $[\mathbf{S}, \mathbf{A}, \mathbf{P}_a, \mathbf{R}_a]$ is composed of state $\mathbf{S}$, action $\mathbf{A}$, transition probability $\mathbf{P}_a$, and reward $\mathbf{R}_a$. The designed random access scheme is mainly based on the Markov decision process.

The state space is determined by the number of MTCDs requesting access to the LEO satellite constellation system.

$$\mathbf{S} = \left[N_1, N_2, ..., N_J\right], \quad (5)$$

N_j is the number of users currently requesting access to LEO satellite j, determined by the type of traffic and satellite coverage time T_j.

Contention-based random access and the EAB scheme [30] are effective approaches for this kind of random access services. When $N > a_{\max}$, the available RAOs are insufficient, whereupon the EAB scheme implements scheduling.

$$C = p_{EAB} \cdot N, \quad (6)$$

where C is the number of contending MTCDs, that are faced with congestion in the contention-based random access control.

The efficiency of random access r_{eff} is defined,

$$r_{eff} = C \cdot \left(\frac{1}{a}\right)\left(1 - \frac{1}{a}\right)^{C-1}, \quad (7)$$

where a indicates the incidence of utilizing RAOs. Furthermore, it suggests the quantity of MTCDs connected successfully in a given resource as well as that of contending MTCDs C, namely the probability of access to the certain MTCDs on the fulfilling of the existing RAOs. If there are now a RAOs and C contending MTCDs, the problem of random access,

$$\begin{aligned} &\max r_{eff} = C \cdot \left(\tfrac{1}{a}\right)\left(1 - \tfrac{1}{a}\right)^{C-1}, \\ &s.t. \ \begin{array}{l} a \in \mathbf{A}, \\ \mathbb{E}\left[D\right] \leq D_{req}, \end{array} \end{aligned} \quad (8)$$

where $\mathbb{E}\left[D\right]$ is the time tolerance on average whenever devices access randomly to RAOs. D_{req} is the maximum of time tolerance for access requests we defined in the context of the paper. $\mathbb{E}\left[D\right]$ can be calculated by the following formula [31],

$$\mathbb{E}\left[D\right] = \sum_{r=0}^{\infty} T_j \cdot (r+1) \cdot p_{access} \cdot (1 - p_{access})^r = \frac{T_j}{p_{access}}. \quad (9)$$

Among them, T_j is the time frame, and p_{access} means that the current MTCDs successfully implements random access in competition. When the EAB scheme executes, the successful access probability p_{access} is,

$$p_{access} = p_{EAB} \cdot p_{succ}, \quad (10)$$

where p_{succ} is the probability of MTCDs successfully connecting with other contending MTCDs. p_{succ} can be calculated as,

$$p_{succ} = a \cdot \left(\frac{1}{a}\right)\left(1 - \frac{1}{a}\right)^{C-1} = \left(1 - \frac{1}{a}\right)^{C-1}. \quad (11)$$

Meanwhile, the next state $N^{(t+1)}$ will change, and idle RAOs can be obtained in the current frame t. The proportion $p_{idle}^{(t)}$ of idle RAOs $a_{idle}^{(t)}$ in the previous $a^{(t)}$ will be obtained:

$$p_{idle}^{(t)} = \left(1 - \frac{1}{a^{(t)}}\right)^{C^{(t)}}. \quad (12)$$

The proportion $p_{idle}^{(t)}$ of idle RAOs can also be calculated by the number of contending MTCDs $C^{(t)}$ and the number of RAOs in the current state $a^{(t)}$:

$$p_{idle}^{(t)} = \frac{a^{(t)} - C^{(t)}}{a^{(t)}}. \quad (13)$$

By calculating the formula (12), the number of contending MTCDs $C^{(t)}$ in the tth state can be estimated as

$$C^{(t)} = \frac{\log\left(p_{idle}^{(t)}\right)}{\log\left(\left(a^{(t)} - 1\right)/a^{(t)}\right)}. \quad (14)$$

The final calculation result of random access MTCDs is as follows,

$$N^{(t+1)} = \frac{C^{(t)}}{p_{EAB}^{(t)}}. \quad (15)$$

Contrary to the above case, when $N^{(t)} \geq a_{\max}$ occurs, the probability of EAB mechanism p_{EAB} needs to be calculated, and calculation of p_{EAB} can be run by r_{eff}. Therefore, it is necessary to process $C^{(t)} = N^{(t+1)} \cdot p_{EAB}^{(t)}$ to get the result of MTCDs participating in the contention, and the situation is the same as $N^{(t)} \leq a_{\max}$.

The values of RAOs available are set as actions of the Markov decision process in the proposed system. It is noteworthy that this action is discrete.

$$\mathbf{A} = [a_1, a_2, a_3, a_4, a_5, a_6]. \tag{16}$$

The reward is to filter out the low-efficiency random access r_{eff} for the current action and state:

$$R_a^{(t+1)} = \begin{cases} 1, r_{reff}^{(t+1)} > r_{reff}^{(t)} \\ 0, r_{reff}^{(t+1)} = r_{reff}^{(t)} \\ -1, r_{reff}^{(t+1)} < r_{reff}^{(t)} \end{cases}. \tag{17}$$

Regarding transition probability $\mathbf{P}_a$, a mathematical model built on previous studies observes the following rules[1] The mathematical model follows Poisson process [32] in the contention-based random access mechanism.

$$P_a^{NN'} = \frac{(\lambda T)^a e^{-\lambda}}{a!}. \tag{18}$$

Accordingly, the contention-free random access mechanism can be explained as [32]:

$$P_a^{NN'} = \frac{60t^2(T-t)^3}{T^6}. \tag{19}$$

To facilitate understanding, a glossary of the mathematics notation involved in the formulation shows in Table I.

TABLE I
EXPLANATION OF ABBREVIATIONS.

Notations	Explanations
T_j	The coverage time of satellite j
N	The number of arrived MTCDs before EAB scheme
C	The number of contending MTCDs after EAB scheme
a_{max}	The maximum of allocated RAOs
r_{eff}	The efficiency of random access

C. Random Access Scheme

Random access can be divided into two types: contention-based random access control and contention-free random access control. Two types of random access schemes are classified according to the set M2M traffic characteristics and the coverage time of the LEO satellites. In the contention-based random access procedure, random access preamble sequences are maximally available, while the others can be used in the contention-free random access procedure.

The M2M traffic is classified by its traffic characteristics, namely asynchronous traffic and synchronous traffic in procession. To be specific, asynchronous traffic represents M2M conventional traffic, and synchronous traffic denotes M2M burst traffic. For averting unnecessary signaling overhead, asynchronous traffic is assigned to correspond to contention-based random access. Spontaneously, synchronous traffic corresponds to contention-free random access.

In the cases of traditional asynchronous and weakly related MTCDs, the interactions with LEO satellites are over the entire range of inclination angles. The terminal device initiates random access uniformly within a time period and routinely applies Poisson distribution to modeling

$$P(N = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \tag{20}$$

MTCDs N awaiting random access to be computed in the Poisson stochastic process.

Based on the formula (8), a feasible solution $\mathbf{M}$ can be obtained,

$$\mathbf{M} = \left\{ p_{EAB} \left| \frac{T_j}{p_{EAB}\left(1-\frac{1}{a_{\max}}\right)^{p_{EAB}N-1}} \leq D_{req}, \right. \right. \\ \left. \left. 0 < p_{EAB} < 1 \right\}, \right. \tag{21}$$

where $a = a_{max}$ is a fixed value. The reason of introducing the EAB scheme to deal with the congestion is that the RAOs are too scarce to support contending MTCDs.

The optimal p_{EAB}^* is calculated by the following formula:

$$p_{EAB}^* = \arg \max_{p_{EAB} \in \mathbf{M}} r_{eff}. \tag{22}$$

If $\mathbf{M}$ was not a feasible solution, the optimal p_{EAB}^* should be defined as:

$$p_{EAB}^* = \frac{a_{\max}}{N}. \tag{23}$$

The formula (22) combined with the formula (23), whereupon the solution to contention-based random access proceeds as follows.

$$p_{EAB}^{(t+1)} = \begin{cases} \arg \max_{p_{EAB} \in \mathbf{M}^{(t+1)}} r_{eff}^{(t+1)} & \mathbf{M}^{(t+1)} \neq \phi \\ a_{\max} / N^{(t+1)} & \mathbf{M}^{(t+1)} = \phi \end{cases}, \\ a^{(t+1)} = a_{\max}. \tag{24}$$

The synchronous and highly-related related MTCDs initiate random access in a centralized manner periodically. There emerges a short-lived upsurge in the communication traffic, tightly followed by a sharp decline. Contention-free

[1]For simplicity of subscript in the formula, N stands for the current state, and it stands for $N(t)$ in some formula's subscript. N' stands for the next state, and it also stands for $N(t+1)$ in some formula's subscript. Action and reward are consistent with the situation of N.

mode provides low-latency services for high priority users in situations such as downlink data arrival, handover or positioning. The contention-free random access control is adopted herein to process the synchronous traffic. That is to say, it is energy-saving option in long propagation delays.

3GPP proposes to approximate a burst M2M service with the assistance of Beta distribution, and that enormous MTCDs simultaneously initiate random access requests at a certain moment. According to the 3GPP TR37.868 [32], the probability density of the reference model for communication traffic that simulates M2M terminals is:

$$f\left(t\right)=\frac{60t^{2}(T_{j}-t)^{3}}{{T_{j}}^{6}}. \tag{25}$$

In this case, the RAOs are sufficient when $N \leq a_{max}$. The access probability p_{EAB} in the EAB access control is set as 1.

Then the optimal number a^{*} of RAOs can be obtained:

$$\begin{aligned} &a^{*}=\arg\max_{a\in\mathbf{L}} r_{eff}, \\ &s.t.\mathbf{L}=\left\{a\left|\frac{T_{j}}{\left(1-\frac{1}{a}\right)^{C-1}}\leq D_{req}, a\in\mathbf{L}\right.\right\}. \end{aligned} \tag{26}$$

If a feasible solution was not available in $\mathbf{L}$, it can be used $a^{*}=C$ could serve as a backup.

Similar to formula (24), the solution to contention-free random access control can be obtained.

$$\begin{aligned} &p_{EAB}^{(t+1)}=1, \\ &a^{(t+1)}=\begin{cases} \arg\max\limits_{a\in\mathbf{L}^{(t+1)}} r_{eff}^{(t+1)} & \mathbf{L}^{(t+1)}\neq\phi \\ \widehat{N}^{(t+1)} & \mathbf{L}^{(t+1)}=\phi \end{cases}. \end{aligned} \tag{27}$$

III. Random Access Control And RAOs Allocation By deep Dyna-Q learning

In this section, the deep Dyna-Q learning algorithm is introduced firstly. Then comes a model-free DRL algorithm to compare in the performance of deep Dyna reinforcement learning algorithm. Finally, it notes a deep Dyna-Q learning algorithm in the system model, by which some latent problems will be addressed.

A. Deep Dyna-Q Learning Algorithm

Reinforcement learning algorithms can be categorized into: model-free method and model-based method [33]. The research on model-free methods is heated because it is not indispensable to train the model but to directly obtain the policy based on the reward function. This method of reducing computational complexity makes it convenient to find an optimal strategy. Model-based reinforcement learning is more complicated because the prerequisites are building a world model and employing dynamic programming based on it.

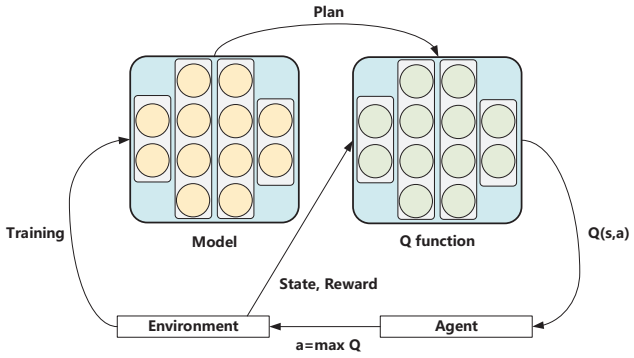

Fig. 2. Deep Dyna-Q learning framework.

However, a suitable and desirable world model free the agents from following the steps and passively waiting for feedback. Especially in the satellites IoT networks, the confounding factors of long distance transmissions and complex atmospheric environments may complicate interactions between terrestrial MTCDs and satellites. A reliable virtual world model proves to pick out the optimal policy by directly predicting the incidence of all the situations. Meanwhile, it enhances sample efficiency rather than over-training with redundant samples in model-free Q-learning.

Model-based reinforcement learning learns from a model of the environment's dynamics. As the following models demonstrate: one is the state transition prediction model $\mathbf{P}$, inputting the current state $s^{(t)}$ and action $a^{(t)}$, and predicting the next state $s^{(t+1)}$. The other is the reward prediction model $\mathbf{R}$, where the operator inputs the current state $s^{(t)}$ and action $a^{(t)}$ to predict the reward $r^{(t+1)}$ of the actual environment. Thus, the world model can be described as follows:

$$s^{(t+1)}\sim\mathbf{P}\left(s^{(t+1)}\left|s^{(t)},a^{(t)}\right.\right), \tag{28}$$

$$r^{(t+1)}\sim\mathbf{R}\left(r^{(t+1)}\left|s^{(t)},a^{(t)}\right.\right). \tag{29}$$

In most instances, model-based reinforcement learning coworks with model-free reinforcement learning. Hence, the framework of Dyna algorithm is a common practice. The Dyna architecture is not a specific reinforcement learning algorithm but a general term for a class of algorithms. The Dyna architecture integrates both methods, learning from the model and the experience of interacting with the environment, thereby updating the value function and policy function. As for the state transition prediction model $\mathbf{P}$ or the reward prediction model $\mathbf{R}$ including value-based iteration and function approximation. The value-based iteration grounded on sample data in the agent's movement can calculate a relatively simple transition, but it requires overloaded

capacity when engaged in a complex environment. The function approximation utilizes the distribution, including the linear model and the Gaussian process [33], to construct the state transition prediction model $\mathbf{P}$ or the reward prediction model $\mathbf{R}$. DNNs also approximate the state transition prediction model $\mathbf{P}$ or the reward prediction model $\mathbf{R}$, with the help of its back propagation approach. DNNs are utilized to approximate the world model in deep Dyna architecture. Incorporation of certain model-free reinforcement learning into the framework of deep Dyna reinforcement learning can generate different algorithms. Within that framework, the deep Dyna-Q learning algorithm is realized via the deep Q-learning.

Such attempts to adopt the sampling training method and build a virtual model through calibration hinge on the presupposition that there is no current-situation-tailored virtual model. However, there is a high likelihood that the method would complicate the problem and lead to the model bias. At this time, DNN can be utilized to train the model to represent the environment. The σ_l is activation function of l layer and $\mathbf{W}_l$ are linear weight functions.

$$\phi(s,\theta) := \sigma_L\left(\mathbf{W}_L\left(\sigma_L\right)\mathbf{W}_{L-1}\left(\ldots\left(\sigma_1\left(\mathbf{W}_1\mathbf{s}\right)\right)\right)\right). \quad (30)$$

After the establishment of the virtual world model, the next steps $\mathbf{S}$ and $\mathbf{A}$ can be accomplished. To generate Q value, DNN is deployed to import the action and state.

If we use deep Q-learning based on the value function, the deep Dyna-Q learning algorithm can be obtained. The world model $(\mathbf{P},\mathbf{R})$ functions in the state transition prediction model $\mathbf{P}$ and the reward prediction model $\mathbf{R}$. As a deep Dyna-Q learning framework is illustrated in Fig. 2, there are two DNNs approximating the world model $(\mathbf{P},\mathbf{R})$ and Q-function. In round of iteration, the Dyna architecture first reacts to the actual environment and updates the state and reward function, then forecasts the world model n times and updates the value function. Both the replay experience of interacting with the environment and the prediction of the world model are engaged in the deep Dyna-Q learning.

B. Solved by Model-Free Deep Q-Learning Network

In the model-free deep Q-learning network, the agent does not resort to the world model to resolve the problem. Inspired by Q-learning, the agents, namely the LEO satellites, react to the present environment in each episode of training [34]. The MTCDs $N(t)$ request for random access in the actual environment. The full utilization of ε- greedy policy, LEO satellite selects the allocated RAOs $a^{(t)}$ to react to the current environment precisely. It also measures $N > a_{max}$ or $N <= a_{max}$ to consider whether take p_{EAB} scheme. If $N <= a_{max}$, the LEO satellite j can respond to requests for random access from MTCDs as many as possible. In this case, the EAB scheme does not have process these requests, i.e., the probability of EAB scheme p_{EAB} equals to 1. Nevertheless, when $N > a_{max}$, the LEO satellite j has to start with its EAB scheme to deal with excessive MTCDs. The arrived MTCDs N in random access will be transformed into contending MTCDs C. Then the state $N^{(t)}$ steps into the next state $N^{(t+1)}$, which derives from the following random numbers in distribution. Subsequently, the LEO satellite j gains the corresponding reward $Reward^{(t+1)}$ via the formula (25). In the process of direct reinforcement learning, the LEO satellite interacts with the actual environment. In each step, the LEO satellite j collects observations regarding the state of the environment and gives response accordingly by utilizing the $\varepsilon-$ greedy policy. Specifically, it selects action a randomly with the probability of ε. Otherwise, it will follow [35]:

$$a^{(t)} = \arg\max_{a} Q\left(N^{(t)}, a; \theta_Q\right), \quad (31)$$

θ_Q are the parameters of the deep Q-learning model, and $Q\left(N^{(t+1)}, a; \theta_Q\right)$ is the approximated value function. Finally, the actual experience D_u consists of $\left(N^{(t)}, a^{(t)}, R^{(t)}, N^{(t+1)}\right)$. The parameters θ_Q in the deep Q-learning model is improved by:

$$L\left(\theta_Q\right) = \mathbb{E}_{\left(N,a,R,N'\right)\sim D_u}\left[\left(y_i - Q\left(N^{(t+1)}, a^{(t+1)}; \theta_Q\right)\right)\right], \quad (32)$$

and the output of deep Q-learning model θ_Q is:

$$y_i = R + \gamma \max_{a^{(t+1)}} Q\left(N^{(t+1)}, a^{(t+1)}; \theta_Q\right), \quad (33)$$

where $\gamma \in [0,1]$ is an attenuation coefficient. $Q(\cdot)$ is the target value updated with each training step.

By minimizing the value of cost function concerning θ_Q, the following gradient can be obtained:

$$\nabla_{\theta_Q} L\left(\theta_Q\right) = \mathbb{E}_{\left(N,a,R,N'\right)\sim D^u}\left[\left(R + \gamma \max_{a'} Q'\left(N', a'; \theta_{Q'}\right) - Q\left(N, a; \theta_Q\right)\right) \cdot \nabla_{\theta_Q} Q\left(s, a; \theta_Q\right)\right]. \quad (34)$$

The utilization of deep Q-learning aims at the development of $Q(\cdot)$ in each iteration. Thus, the pseudo code of the model-free deep Q-learning algorithm for random access is concluded in Algorithm 1.

C. Solved by Deep Dyna-Q Learning Network

Of two DNNs in the deep Dyna-Q learning algorithm, one approximates the Q-function, and the other stands for the world model. Deep Dyna-Q learning algorithm can be split into three phases [36]. In the phase one, the LEO satellite j utilizes direct reinforcement learning to gain actual experience of D_u and promotes the Q-function model. In the phase two, the world model $(\mathbf{P},\mathbf{R})$ is trained in actual experience D_u and gains simulated experience D_u in the meantime. In the phase three, the LEO satellite j takes

Algorithm 1 Model-Free Deep Q-Learning Random Access in Satellites IoT Networks

Initialize: All parameters θ_Q of Q-network are generated from truncated normal distribution randomly, and the value Q corresponding to all states and actions is initialized based on $\mathbf{\Theta}$. Clear the set D_u of experience replay.
for i from 1 to T_j:
a) Initialize the number of arrived MTCDs N_0 as the first state of the current states;
b) In the DQN, input N and output Q values of different RAOs;
c) Use $\varepsilon - greedy$ method: with probability ε select a random allocated RAO a, or select $a^{(t)} = \arg\max_a Q(N, a; \theta_Q)$;
d) Execute selected RAO a, and observe the number of arrived MTCDs N and efficiency of random access r_{eff};
e) **If** $N > a_{max}$:
1) Compute probability p_{EAB} in EAB scheme in formula (15);
2) Compute the number of contending MTCDs $C = p_{EAB} \times N$;
Else:
1) Compute the number of contending MTCDs $C = N$;
f) Compute efficiency of random access r_{eff};
g) Update the next number of arrived MTCDs N';
h) Store (N, a, R, N') to replay experience D_u;
i) Randomly sample with batch size of them from the actual experience buffer D_u.
end for
Output: The trained model-free deep Q-learning with parameters θ_Q.

Algorithm 2 Deep Dyna-Q Learning Network Random Access in Satellite IoT Networks

Initialize: All parameters θ_Q of Q-network and θ_M of world model are generated from truncated normal distribution randomly, and the value Q corresponding to all states and actions is initialized based on $\mathbf{\Theta}$. Clear the set D_u of actual experience replay and D_s of stimulated experience replay.
for i from 1 to Iteration:
Direct Reinforcement Learning starts
a) It executes step a) to i) in Algorithm 1.
Direct Reinforcement Learning ends
World Model Learning starts
a) Randomly sampling in minibatches among the trained samples (N, a, R, N') from experience replay D_u;
b) Update θ_M via minibatch SGD.
World Model Learning ends
Planning starts
for k from 1 to K:
1) LEO satellite selects a random-access action a with probability ε, or selects $a^{(t)} = \arg\max_a Q(N, a; \theta_Q)$;
2) World model responds with reward R and the next state N';
3) Update state to N';
4) Store (N, a, R, N') to stimulated experience replay D_s.
end for
Planning ends
end for
Output: The trained model-free deep Q-learning with parameters θ_Q and the trained world model with parameters θ_M.

advantages of simulated experience D_u. to implement the random access policy, which is under continual improvement.

In the phase of direct reinforcement learning, the agent interacts with the real environment. A solid description of this process is displayed in the Section II.B, and direct reinforcement learning utilizes model-free deep Q-learning to train the DQN.

In this phase of world model learning, the world model $(\mathbf{P}, \mathbf{R})$ is parameterize as $M(N, a; \theta_M)$. The input of world model is the current state, i.e., the arrived MTCDs N and the allocated RAOs a.

Similar to formula (32), SGD is deployed to train the world model $M(N, a; \theta_M)$

$$L(\theta_M) = \mathbb{E}_{(N,a,R,N')\sim D_s}\left[\left(y_i - Q\left(N^{(t+1)}, a^{(t+1)}; \theta_M\right)\right)\right]. \tag{35}$$

In the planning process, the established world model $(\mathbf{P}, \mathbf{R})$ facilitates the LEO satellite j in making decisions. Supposing that the world model can simulate the environment accurately, a reasonable planning step K contributes to the LEO satellite in learning policies to interact with the arrived MTCDs from simulated experience D_u.

$$L(\theta_Q) = \mathbb{E}_{(N,a,R,N')\sim D_u}\left[\left(y_i - Q\left(N^{(t+1)}, a^{(t+1)}; \theta_Q\right)\right)\right]. \tag{36}$$

The optimal paths for world model and DQN in deep Dyna-Q learning echoes the training principle of model-free deep Q-learning in formula (34). Thus, the pseudo code of the deep Dyna-Q learning algorithm for random access is concluded in Algorithm 2.

IV. Simulation Results And Discussions

This section verifies the performance of the proposed control algorithms for random access in the LEO satellite IoT networks, considering contention-based access and contention-free access. The simulation parameters of two typical traffic modes are listed in Table II. In simulation, there are three latent layers hidden in the world model within the

TABLE II
TWO DIFFERENT RANDOM ACCESS SIMULATION PARAMETERS.

Traffic Model	Synchronous	Asynchronous
Traffic Distribution	Beta Distribution(3,4)	Poisson Distribution λ
Period Time	10 seconds	60 seconds
Frequency Multiplexed Factor	$F = 8$	$F = 1$
Random Access Mode	Contention-Free	Contention-Based

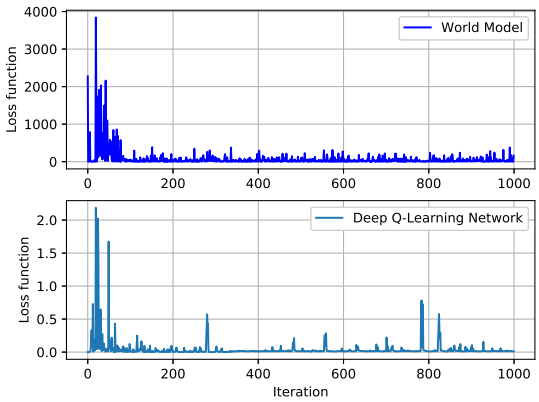

Fig. 3. The loss function of world model in deep dyna-Q learning via iterations

framework of the deep Dyna-Q learning. The output layer outputs two parameters, including predicting the state and the corresponding reward. The learning rate of each DNN is 1×10^{-3}, and the optimizer is deployed as Adam. Q value function has two hidden layers, and its sizes are 16. The learning rate is 1×10^{-3}. The batch sizes of actual experience D_u and stimulated experience D_s are 32 whose replay memory is 10000. According to the coverage time of each LEO satellite, the orbital altitude of LEO satellite h_j is set as 700 km. The latitude at projection of the LEO satellite j on the earth's surface α_j is generated from the uniform distribution between $0^\circ - 60^\circ$ and elevation angle θ_j derives from the uniform distribution between $0^\circ - 90^\circ$. The process environment is deployed in PyTorch v1.4.0.

Fig. 3 depicts the loss function for the proposed deep Dyna-Q learning algorithm with the training iterations increasing. In training the world model, each DNN is established with one hidden layer in a size of 50. Additionally, each DNN is given with a ReLU activation function. As is shown in Fig. 3, the loss function for the virtual world model converges quickly when the iteration step reaches 600. Thus, the following discussion mainly focuses on the trained world model after 400 iteration steps.

Fig. 4 shows the total reward of deep Dyna-Q learning with increasing training episodes for different training steps

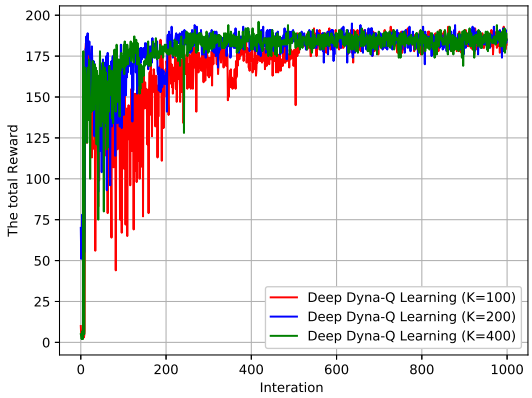

Fig. 4. The reward function for Deep Dyna-Q Learning in different training steps

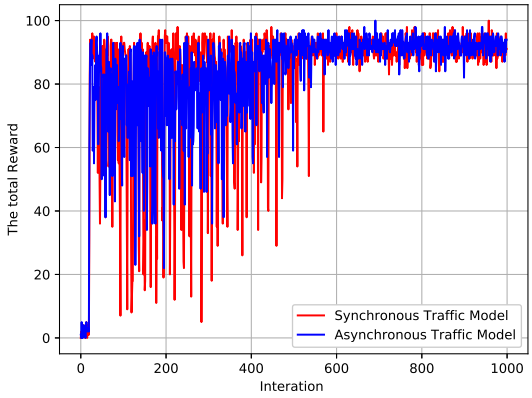

Fig. 5. The reward function with different methods for the asynchronous traffic model.

K. The simulation settings of learning rate of world model and DQN, arrive rate λ, and a_{max} are the same as in Fig. 3. The simulation results verify the significance of planning steps, and it reflects that the trained world model can contribute to learning. And when the training step $K = 400$, it will better convergence than $K = 100$ and $K = 200$. This is because the good virtual model of deep Dyna-Q learning can accelerate to obtain the maximum value of the reward function with planning.

Fig. 5 illustrates the training results between the synchronous traffic model and the asynchronous traffic model. Deep Q-learning is utilized to observe the convergence of two different traffic models. In Fig. 5, the training steps are set as 100. It suggests that the asynchronous traffic model outperforms the synchronous one in readiness and stability. The characteristics of asynchronous traffic model,

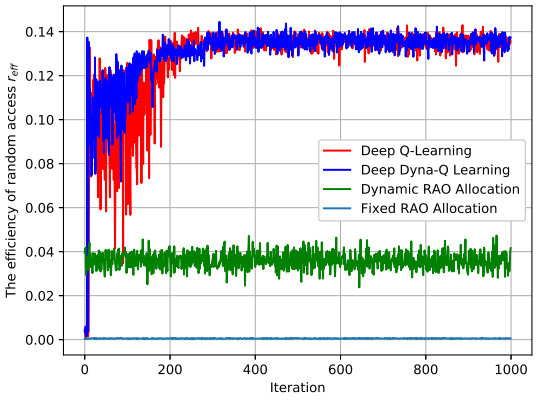

Fig. 6. The efficiency of random access versus iterations.

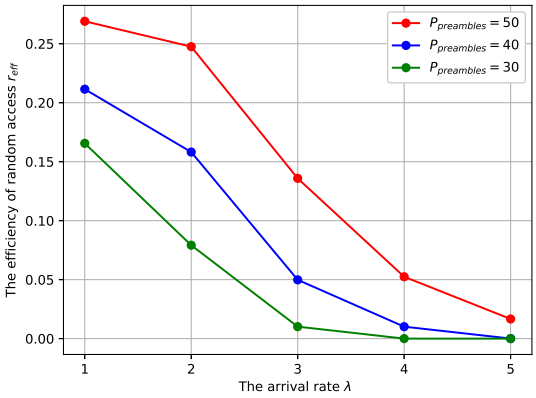

Fig. 7. The efficiency of random access versus different arrival rate.

being traditional and weakly-related, enable to accelerate the learning in the proposed schemes.

Fig. 6 depicts the convergence of efficiency R_{eff} between deep Dyna-Q Learning and deep Q-Learning. Compared with deep Q-Learning, R_{eff} in deep Dyna-Q Learning tends to be more stable and swift in convergence. The arrival rate λ is 3 and the maximum of available preamble sequences is set as 54. Moreover, Dynamic RAO allocation and fixed RAO allocation are introduced in Fig. 6. It reflects the advantages of the two proposed algorithms.

Fig. 7 shows the efficiency of random access R_{eff} when the arrival rate λ increases from 1 to 5, especially when $P_{preambles} = 30$, $P_{preambles} = 40$, or $P_{preambles} = 50$. R_{eff} decreases as the arrival rate λ increases in the traffic model 1. It illustrates the random access capacity of the proposed random-access scheme attenuates when the density of MTCDs increases. Moreover, a larger maximum of preamble sequences $P_{preambles}$ results in a higher efficiency of random access R_{eff}.

V. Conclusion

This paper researches a deep Dyna reinforcement learning scheme for random access control in LEO satellite IoT networks. Firstly, the random access control in LEO satellite IoT networks is formulated based on the Markov decision process. Then, contention-based random access and contention-free random access are explored in the dimensions of M2M traffic characteristics and the coverage of LEO satellites. In the settings of RAOs and limited delay, it proposes the control model for random access via the maximum efficiency of random access. The model-free deep Q-learning network algorithm aims solve the problem by means of the random access model. Subsequently, the deep Dyna-Q learning algorithm is applied to the designed control model for random access. To be precise, the agent improved the random access model-free policy by dint of simulation. Since the proposed deep Dyna-Q learning scheme is based on the established world model, it presents performance similar to that of the model-free scheme with low involvement of the entire LEO satellite IoT networks.

References

[1] W. Zhan and L. Dai, "Massive random access of machine-to-machine communications in LTE networks: Modeling and throughput optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2771–2785, Apr. 2018.

[2] B. Di, H. Zhang, L. Song, Y. Li, and G. Y. Li, "Ultra-dense LEO: Integrating terrestrial-satellite networks into 5G and beyond for data offloading," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 47–62, Jan. 2019.

[3] D. Chen et al., "Resource cube: Multi-virtual resource management for integrated satellite-terrestrial industrial IoT networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11963–11974, Oct. 2020.

[4] S. Lien, T. Liau, C. Kao, and K. Chen, "Cooperative access class barring for machine-to-machine communications," *IEEE Trans. Wireless Commun.*, vol. 11, no. 11, pp. 27–32, Jan. 2012.

[5] S. Y. Lien, K. C. Chen, and Y. Lin, "Toward ubiquitous massive accesses in 3GPP machine-to-machine communications," *IEEE Commun. Mag.*, vol. 49, no. 4, 2011.

[6] C. Y. Oh, D. Hwang, and T. J. Lee, "Joint access control and resource allocation for concurrent and massive access of M2M devices," *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4182–4192, Aug. 2015.

[7] Y. Mehmood, N. Haider, M. Imran, A. Timm. Giel, and M. Guizani, "M2M communications in 5G: State-of-the-art architecture, recent advances, and research challenges," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 194–201, Sept. 2017.

[8] G. Choudhury and S. Rappaport, "Diversity ALOHA-A random access scheme for satellite communications," *IEEE Trans. Commun.*, vol. 31, no. 3, pp. 450–457, Mar. 1983.

[9] M. Bacco et al., "IoT applications and services in space information networks," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 31–37, Apr. 2019.

[10] O. del Rio Herrero and R. De Gaudenzi, "Generalized analytical framework for the performance assessment of slotted random access protocols," *IEEE Trans. Wireless Commun.*, vol. 13, no. 2, pp. 809–821, Feb. 2014.

[11] M. Lee, J. Lee, J. Lee, and J. Lim, "R-CRDSA: Reservation-contention resolution diversity slotted ALOHA for satellite networks," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1576–1579, Oct. 2012.

[12] R. De Gaudenzi, O. del Rio Herrero, G. Acar, and E. Garrido Barrabes, "Asynchronous contention resolution diversity ALOHA: Making CRDSA truly asynchronous," *IEEE Trans. Wireless Commun.*, vol. 13, no.11, pp. 6193–6206, Nov. 2014.

[13] Y. Sim and D. H. Cho, "Performance analysis of priority-based access class barring scheme for massive MTC random access," *IEEE Syst. J.*, vol. 14, no. 4, pp. 5245–5252, Dec. 2020.

[14] P. K. Wali and D. Das, "Optimization of barring factor enabled extended access barring for energy efficiency in LTE-advanced base station," *IEEE Trans. Green Commun. Networking*, vol. 2, no. 3, pp. 830–843, Sept. 2018.

[15] M. Bacco, P. Cassar, M. Colucci, and A. Gotta, "Modeling reliable M2M/IoT traffic over random access satellite links in non-saturated conditions," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 5, pp. 1042–1051, May 2018.

[16] M. Bacco, T. De Cola, G. Giambene, and A. Gotta, "TCP-based M2M traffic via random-access satellite links: Throughput estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 2, pp. 846–863, Apr. 2019.

[17] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2224–2287, May 2019.

[18] Z. Lin, M. Lin, J. Wang, T. de Cola, and J. Wang, "Joint beamforming and power allocation for satellite-terrestrial integrated networks with non-orthogonal multiple access," *IEEE J. Sel. Top. Signal Process.*, vol. 13, no. 3, pp. 657–670, Jun. 2019.

[19] W. Usaha and J. A. Barria, "Reinforcement learning for resource allocation in LEO satellite networks," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 37, no. 3, pp. 515–527, Jun. 2007.

[20] H. Huang, S. Guo, W. Liang, K. Wang, and A. Y. Zomaya, "Green data-collection from geo-distributed IoT networks through low-earth-orbit satellites," *IEEE Trans. Green Commun. Networking*, vol. 3, no. 3, pp. 806–816, Sept. 2019.

[21] F. Rinaldi et al., "Broadcasting services over 5G NR enabled multi-beam non-terrestrial networks," *IEEE Trans. Broadcast.*, pp. 1–13, Jun. 2020.

[22] H. Zhang, N. Yang, W. Huangfu, K. Long, and V. C. M. Leung, "Power control based on deep reinforcement learning for spectrum sharing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4209–4219, Jun. 2020.

[23] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Trans. Cognit. Commun. Networking.*, vol. 4, no. 2, pp. 257–265, Jun. 2018.

[24] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, Jan. 2019.

[25] C. Zhong, Z. Lu, M. C. Gursoy, and S. Velipasalar, "A deep actor-critic reinforcement learning framework for dynamic multichannel access," *IEEE Trans. Cognit. Commun. Networking.*, vol. 5, no. 4, pp. 1125–1139, Dec. 2019.

[26] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks" *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1277–1290, Jun. 2019.

[27] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[28] F. Meng, P. Chen, L. Wu, and J. Cheng, "Power allocation in multi-user cellular networks: Deep reinforcement learning approaches," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6255–6267, Oct. 2020.

[29] B. Yang, F. He, J. Jin, and X. Guanghan, "Analysis of coverage time and handoff number on LEO satellite communication systems," *J. Electron. Inf. Technol.*, vol. 36, no. 4, pp. 804–809, 2014.

[30] M. S. Ali, E. Hossain, and D. I. Kim, "LTE/LTE-A random access for massive machine-type communications in smart cities," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 76–83, Jan. 2017.

[31] T. P. C. De Andrade, C. A. Astudillo, L. R. Sekijima, and N. L. S. Da Fonseca, "The random access procedure in long term evolution networks for the Internet of Things," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 124–131, Mar. 2017.

[32] 3GPP, "TR37.868 study on RAN improvements for machine type communications."

[33] Sutton,S. Richard, Barto, and G. Andrew, "Reinforcement learning: An introduction (2ed.)," *MIT Press.*, 2008.

[34] C. Qiu, H. Yao, F. R. Yu, F. Xu, and C. Zhao, "Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5871–5883, Jun. 2019.

[35] S. Y. Lien, K. C. Chen, and Y. Lin, "Deep Q-learning-based node positioning for throughput-optimal communications in dynamic UAV swarm network," *IEEE Trans. Cognit. Commun. Networking*, vol. 5, no. 3, pp. 554-566, Sept. 2019.

[36] B. Peng et al., "Deep dyna-q: Integrating planning for task-completion dialogue policy learning,". *arXiv preprint arXiv:1801.06176*, 2018.